When a user (client 106[1]) requests the CNBC home page "www.cnbc.com", with reference to Figures 2 and 3 of the application, the user's browser is provided with the URL "www.cnbc.com"[2]. The browser extracts the host (origin server/content provider server) name from the resource identifier, in this case "www.cnbc.com", and uses a domain name server (DNS) to look up the network (IP) address of the corresponding server[3]. The DNS returns the IP address, in this example, "64.14.56.75". The user's browser then uses that IP address to connect to "www.cnbc.com"[4], and sends a request over the connection identifying the requested resource ("www.cnbc.com")[5].

With reference to Figure 3 of the application, the request for "www.cnbc.com" made by the user is intercepted by the reflector 108[6]. (For the sake of this example, assume that the

---

1 "Client **106** is a processor requesting resources from origin server **102** on behalf of an end user. The client **106** is typically a user agent (e.g., a Web browser such as Netscape Communications Corporation's Navigator™) or a proxy for a user agent." *Specification*, pg. 7, lines 10-13.

2 "A1. A browser (e.g., Netscape's Navigator) at the client receives a resource identifier (i.e., a URL) from a user." *Specification*, pg. 11, lines 10-11.

3 "A2. The browser extracts the host (origin server) name from the resource identifier, and uses a domain name server (DNS) to look up the network (IP) address of the corresponding server. The browser also extracts a port number, if one is present, or uses a default port number (the default port number for http requests is 80)." *Specification*, pg. 11, lines 13-17.

4 "A3. The browser uses the server's network address and port number to establish a connection between the client **106** and the host or origin server **102**." *Specification*, pg. 11, lines 19-21.

5 "A4. The client **106** then sends a (GET) request over the connection identifying the requested resource." *Specification*, pg. 11, lines 23-24.

6 "In this invention reflector **108** effectively takes the place of an ordinary Web server or origin server **102**. The reflector **108** does this by taking over the origin server's IP address and port number. In this way, when a client tries to connect to the origin server **102**, it will actually connect to the reflector **108**. The original Web server (or origin server **102**) must then accept requests at a different network (IP) address, or at the same IP address but on a different port number. Thus, using this invention, the server referred to in A3-A7 above is actually a reflector **108**." *Specification*, pg. 12, lines 17-23.

reflector 108 and the origin server 102 are co-located[7] as shown in Figure 1.) The reflector then analyzes the request and determines whether or not to reflect the request[8]. In this case, e.g., the reflector decides not to reflect the request, but to serve it locally at the origin server, www.cnbc.com. Therefore, in this case, the origin server "www.cnbc.com" process the request[9].

The origin server then receives the request (for the resource "www.cnbc.com"), locates or composes the resource and sends the reply with the resource (i.e., the web page at "www.cnbc.com") to the client[10]. The reflector intercepts the origin server's reply and determines whether or not the reply contains embedded objects[11]. In this case, i.e., in the case of "www.cnbc.com", the reply is an HTML document which includes at least one embedded object, namely "www.cnbc.com/imagestopnav/logo.GIF".

---

7 "The reflector **108** is a mechanism, preferably a software program, that intercepts requests that would normally be sent directly to the origin server **102**. While shown in the drawings as separate components, the reflector **108** and the origin server **102** are typically co-located, e.g , on a particular system such as data server **112**. . . . the reflector **108** may even be a "plug in" module that becomes part of the origin server **102**." *Specification*, pg. 8, lines 13-18.

8 "B1 The reflector **108** analyzes the request to determine whether or not to reflect the request. . . .
B2 If the request is not from a repeater, . . . the reflector either reflects the request . . . or serves the request locally" *Specification*, pg. 13, lines 3-18.

9 "B4. To serve a request locally, the request is sent by the reflector to ("forwarded to") the origin server **102**. In this mode, the reflector acts as a reverse proxy server. The origin server **102** processes the request in the normal manner (A5-A7)."

10 "The origin server **102** processes the request in the normal manner (A5-A7)." *Specification*, pg. 15, lines 8-9.

11 "The reflector then obtains the origin server's reply to the request which it inspects to determine if the requested resource is an HTML document, i.e., whether the requested resource is one which itself contains resource identifiers." *Specification*, pg. 15, lines 9-13.

The resource (i.e., the HTML page from "www.cnbc.com') is then rewritten with modified URLs[12]. The modified HTML page, as modified by rewriting (described below), is then returned to the requesting client **106**. The page, as modified, is served from the origin server.[13]

Rewriting of the HTML page "www.cnbc.com" is performed as follows.[14] For each URL encountered in the resource to be re-written, if the URL is repeatable, it is modified to refer to a selected repeater.[15] Thus, in this example, if the repeater is "fp.cnbc.com", then the URL for the embedded object "www.cnbc.com/imagestopnav/logo.GIF" is changed[16] to:

"fp.cnbc.com/www.cnbc.com/imagestopnav/logo.GIF".

So, the user (client) receives the page from "www.cnbc.com" with at least one embedded object's URL rewritten to refer to a different server ("fp.cnbc.com"). The client's browser then begins to load all the embedded objects in the page. When it tries to load the resource "fp.cnbc.com/www.cnbc.com/imagestopnav/logo.GIF", with reference to Figure 2

---

12 "B5. If the resource is an HTML document then the reflector rewrites the HTML document by modifying resource identifiers (URLs) within it . . ." *Specification*, pg. 15, lines 14-15.

13 "The resource, possibly as modified by rewriting, is then returned in a reply to the requesting client **106**." *Specification*, pg. 15, lines 16-17.

14 "**Rewriting HTML Resources** . . . As explained . . . with reference to FIGURE 3 (B5), when a . . . serves a resource which itself includes resource identifiers (e.g., a HTML resource), that resource is modified (rewritten) to pre-reflect resource identifiers (URLs) of repeatable resources that appear in the resource." *Specification*, pg. 30, lines 4-7.

15 "F3. For each URL encountered in the resource to be re-written, the rewriter must determine whether the URL is repeatable . . .. If . . . the URL is repeatable, it is modified to refer to the selected repeater." *Specification*, pg. 31, lines 13-16.

16 "Given a repeater name, scheme, origin server name and path, create a new URL . . . [using] the following format: *http://<repeater>/<server>/<path>*" *Specification*, pg. 14, lines 13-16.

of the application, it extracts the server name ("fp.cnbc.com") from the URL and obtains an IP address for a server (in this case one of the repeaters) (at A2). Then the client/user, establishes a connection with the server (at A3) and sends a request for the embedded resource "logo.GIF" to that server.

B. DISCUSSION

Please note that some of the terms used in the present application differ from those used in the '703 Patent. However, these terms denote the same or patentably indistinct items. Specifically, the present application uses the term "repeater"[17] or "repeater server" whereas the '703 patent uses the term "content server", and the present application uses the term "origin server"[18] whereas the '703 patent uses the term "content provider server".

The following tables summarize support for two of the claims/counts in the present application.

---

17 ". . . partially replicated servers or repeaters **104a**, **104b**, and **104c**. Each repeater **104a**, **104b**, and **104c** replicates some or all of the information available on the origin server **102** as well as information available on other origin servers in the network **100**" *Specification*, pg. 6, lines 15-18.

18 "Origin server **102** is a server at which resources originate. More generally, the origin server **102** is any process or collection of processes that provide resources in response to requests from a client **106**. Origin server **102** can be any off-the-shelf Web server. In a preferred embodiment, origin server **102** is typically a Web server such as the Apache server or Netscape Communications Corporation's Enterprise™ server." *Specification*, pg. 7, lines 5-9.